



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/613,166

07/03/2003

Adam K. Kolawa

50283/P396

9159

23363 7590 10/03/2008
CHRISTIE, PARKER & HALE, LLP
PO BOX 7068
PASADENA, CA 91109-7068

EXAMINER

TECKLU, ISAAC TUKU

ART UNIT

PAPER NUMBER

2192

MAIL DATE

DELIVERY MODE

10/03/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/613,166	Applicant(s) KOLAWA ET AL.	
	Examiner ISAAC T. TECKLU	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 03 July 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-31 and 43-48 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-31 and 43-48 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the amendment 07/03/2008.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 07/03/2008 has been entered.

Specification

3. The disclosure is objected to because of the following informalities: The specification is devoid of terms such as "verification programs" as recited in claims 1-31 and 43-48. The specification is inconsistent with terms recited in claims 1-31 and 43-48. For example the specification discloses verification tools (page 3, lines 10-15). The specification should be written in "full, clear, concise, and exact terms". Appropriate correction is required.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claims 1-26 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The claims recite “executing a first software verification program relating to a first lifecycle phase of the computer software, wherein the first software verification program automatically generates one or more test cases from the source code of the computer software; executing a second software verification program relating to a second lifecycle phase of the computer software different from the first lifecycle phase, wherein the second software verification program automatically generates one or more test cases from the source code of the computer software” (e.g. claim 1, lines 8-14). The specification describes executing the plurality of software verification tools and automatically generates the test cases (page 3, lines 10-15). Furthermore, the specification describes the capability of executing these verification tools by any of the developers in the entire code without having to generate any test cases (page 14, line 14-18). However, no where in the specification does provide clear explanation as to how the software verification program is related to the “first lifecycle phase” and “second lifecycle phase”. Contrary to the recited claim limitation, the specification describes executing the

Art Unit: 2192

verification tools for the entire software, without specifically directing the related respective lifecycle phases, such as first lifecycle and second lifecycle. In fact, the specification does not describe the first and second lifecycle phases.

For the purpose of examination, examiner has not consider the claim limitation “executing a first software verification program relating to a first lifecycle phase of the computer software, wherein the first software verification program automatically generates one or more test cases from the source code of the computer software; executing a second software verification program relating to a second lifecycle phase of the computer software different from the first lifecycle phase, wherein the second software verification program automatically generates one or more test cases from the source code of the computer software”.

Dependent claims 2-10 and 15-23 are rejected for not remedying to the indefinite teaching in the base claims.

Response to Arguments

6. Applicant's arguments filed 07/03/08 have been fully considered but they are not persuasive.

a) The Applicant asserted: “Amended independent claims 1 and 14 include, among other elements, ‘executing a first software verification program relating to a first lifecycle phase of the computer software, wherein the first software verification program automatically generates one or more test cases from the source code of the computer software; executing a second software verification program relating to a second lifecycle phase of the computer

software different from the first lifecycle phase, wherein the second software verification program automatically generates one or more test cases from the source code of the computer software.’ Jorapur does not teach the above element (page 10).

The examiner would like to direct the Applicant to the above claim rejection under 35 U.S.C 112, 1st paragraph. Furthermore, the examiner respectfully disagrees with the Applicant’s assertion. Jorapur teaches “testing may be ongoing, such that the process of performing the test or plurality of tests (plurality of verification programs) may be automated. Errors or failures may be detected and reported as part of the testing framework at any stages (lifecycle phases). During deployment, execution, and undeployment, testing may include verifying the configuration and/or operation of individual modules of the application to provide more precise identification of errors or failures, for example using code inserted into the application and/or its modules” (col. 10:10-30 - emphasis added). Furthermore, Jorapur teaches executing different tests (verification programs) relating to different lifecycle phase of the computer software (col. 7:140-54, col7:55-65, col. 825-36 and for example in FIG. 4, steps 401-410). Therefore Jorapur teaches detecting and analyzing an error at any stage or lifecycle phase. Accordingly, Applicant’s argument is not persuasive.

b) The Applicant asserted: “Neither these individual tests 302, nor the test generator 301 of Jorapur can be constituted as the claimed “plurality of software verification programs,” for example, JtestTM, C++TestTM (page 11).

The examiner would like to reiterate that it is noted that the features upon which applicant relies (i.e., JtestTM, C++TestTM) are not recited in the rejected claim(s). Jorapur generates a plurality of tests to test a software program application. Again, the plain language of the claim merely recites “plurality of software verification programs” and Jorapur plurality of tests are tools/programs used to test software to verify the software error at any stage (lifecycle phase), as noted above (col. 10:10-30).

c) The Applicant asserted: “software verification program is different than simply a test that may be executed by the program (or the tool) among other tests. (page 11).

The examiner respectfully disagrees with Applicant’s assertions. Jorapur generates a plurality of tests to test of software. The plain language of the claim merely recites “plurality of software verification programs” and Jorapur plurality of tests are tools/programs used to test software verify the software error at any stage (lifecycle phase). As broadly speaking any prior art that teaches plurality of tests would read on the limitation “plurality of software verification programs” because all that is required by this limitation is that the fact that the software verification programs tests the software for errors. Applicant is reminded that claims are given their broadest reasonable interpretation in light of the supporting disclosure. *In re Morris*, 127 F.3d 1048, 1054-55, 44 USPQ2d 1023, 1027-28 (Fed. Cir.1997). Limitations appearing in the specification but not recited in the claim should not be read into the claim. *E-Pass Techs., Inc. v.3Com Corp.*, 343 F.3d 1364, 1369, 67 USPQ2d 1947, 1950 (Fed. Cir. 2003) (claims must be interpreted "in view of the specification" without importing limitations from the specification

Art Unit: 2192

into the claims unnecessarily). *In re Prater*, 415 F.2d 1393, 1404-05, 162 USPQ 541,550- 551 (CCPA 1969).

d) The Applicant asserted: “No known error is provided in any of the test performed by Jorapur’s method” (page 12).

The examiner respectfully disagrees with the Applicant’s assertion. Jorapur teaches “executing a test may involve executing the application within a test framework according to one or more test configurations and detecting errors occurring during the application execution (emphasis added). The framework may provide an execution environment for the application to be tested. For example, it may provide services to the software, simulate or provide resources, and/or may provide stability so that errors in the software have a limited impact on other software then executing or ensure that the tests/software terminate gracefully” (col. 3:10-25 - emphasis added). Accordingly, Applicant’s argument is no persuasive.

e) The Applicant asserted: “determining what phase of the life cycle the error was introduced, based on analyzing the known error” (page 13).

The examiner respectfully disagrees with the Applicant’s assertion. Jorapur teaches “testing may be ongoing, such that the process of performing the test or plurality of tests may be automated. Errors or failures may be detected and reported as part of the testing framework at any stage. During deployment, execution, and undeployment, testing may include verifying the configuration and/or operation of individual modules of the application to provide more precise identification of errors or failures, for example using code inserted into the application and/or its

Art Unit: 2192

modules” (col. 10:10-30 - emphasis added). Therefore Jorapur teaches detecting and analyzing an error at any stage or lifecycle. Accordingly, Applicant’s argument is not persuasive.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 1-5, 7-18, 20-30 and 43-48 are rejected under 35 U.S.C. 102(e) as being anticipated by Jorapur (US 7,299,382 B2).

Per claim 1 (Currently Amended)

Jorapur discloses a method for automatically preventing errors in computer software having a plurality of different life cycle phases, the method comprising:

storing source code of the computer software in a code repository (e.g. FIG. 4, step 401 and FIG. 5, 501 and related text).

Art Unit: 2192

~~executing~~ providing a plurality of software verification ~~tools~~ programs to verify the computer software (col. 4:55-65 "... each test may be generated in one or more blocks corresponding to one or more parts of the application to be tested ..." and col. 6: 52-60 "... tests 302 generated may include test code that may be inserted ..." and col. 9: 45-55 "... test may be executed to assess the operation and function of the application to be tested ..." and e.g. FIG. 4, step 408 and related text), wherein each of the plurality of software verification ~~tools~~ corresponds programs relates to a respective lifecycle phase of the computer software and automatically generates one or more test cases from the source code of the computer software (col. 11:30-35 "... produce multiple test cases ...");

executing a first software verification program relating to a first lifecycle phase of the computer software, wherein the first software verification program automatically generates one or more test cases from the source code of the computer software (col. 10:10-30 "...Errors or failures may be detected and reported as part of the testing framework at any stage. During deployment, execution, and undeployment, testing may include verifying the configuration and/or operation of individual modules of the application to provide more precise identification of errors or failures);

executing a second software verification program relating to a second lifecycle phase of the computer software different from the first lifecycle phase, wherein the second software verification program automatically generates one or more test cases from the source code of the computer software (col. 10:10-30 "...Errors or failures may be detected and reported as part of the testing framework at any stage. During deployment, execution, and undeployment, testing

Art Unit: 2192

may include verifying the configuration and/or operation of individual modules of the application to provide more precise identification of errors or failures);

generating verification results for each respective lifecycle phase of the computer software, responsive to executing the first and second software verification programs ~~plurality of software verification tools~~ and the automatically generated test cases (col. 14:20-25 "... results may be gathered after generation ..." and col. 11:30-35 "... produce multiple test cases ...") and;

processing the verification results for generating a representation of functional behavior the computer software (col. 3:10-25 "... results may reflect some behavior of the application during execution ... results may reflect operations during deployment and un deployment of the application to be tested ...");

Per claim 2 (Currently Amended)

Jorapur discloses:

The method of claim 1 further comprising providing a common configuration file for the plurality of verification ~~tools~~ programs (e.g. FIG. 3, Configurations 303 and related text).

Per claim 3 (Currently Amended)

Jorapur discloses:

The method of claim 2, further comprising customizing a verification scope of one or more of the verification ~~tools~~ programs by modifying the common configuration file responsive

Art Unit: 2192

to an objective criterion of quality of the computer software (col. 11:40-50 "... different attributes may be specified in a configuration file ..." and col. 10:1-15 "... configurations may be changed ...").

Per claim 4 (Currently Amended)

Jorapur discloses:

The method of claim 3 further comprising modifying a portion of the common configuration file specific to one of the plurality of verification tools responsive to the objective criterion of quality of the computer software (col. 10:1-15 "... configurations may be changed ..." and e.g. FIG. 4, step 407 and related text).

Per claim 5 (Currently Amended)

Jorapur discloses:

The method of claim 3 further comprising modifying a portion of the common configuration file specific to one of a plurality of software developers responsive to the objective criterion of quality of the computer software (col. 10:1-15 "... configurations may be changed ...").

Per claim 7

Jorapur discloses:

The method of claim 1, wherein each portion of the computer software being developed by a software developer of a plurality of software developers, and the verification results include a plurality of objective criteria each of the plurality of objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer of the plurality of software developers (col. 3:10-25 "... results may reflect some behavior of the application during execution ... results may reflect operations during deployment and un deployment of the application to be tested ...").

Per claim 8 (Currently Amended)

Jorapur discloses:

The method of claim 7 further comprising providing a common configuration file for the plurality of verification ~~tools~~ programs; and modifying the common configuration file responsive to one or more objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer (col. 10:1-15 "... configurations may be changed ...").

Per claim 9 (Currently Amended)

Jorapur discloses:

The method of claim 7 further comprising verifying a first portion of the computer software developed by a first developer of the plurality of software developers using the plurality of verification ~~tools~~ programs, before the computer software is stored in the code

Art Unit: 2192

repository (col. 14:20-25 “... results may be gathered after generation ...” and col. 11:30-35 “... produce multiple test cases ...”).

Per claim 10

Jorapur discloses:

The method of claim 9 further comprising allowing storing the first portion of the computer software in the code repository only if result of verification of the first portion meets a set standard (col. 14:20-25 “... results may be gathered after generation ...” and col. 11:30-35 “... produce multiple test cases ...”).

Per claim 11

Jorapur discloses:

The method of claim 10 further comprising modifying the set standard responsive to the objective criterion of quality of the computer software (col. 10:1-15 “... configurations may be changed ...”).

Per claim 12

Jorapur discloses:

The method of claim 10, wherein the set standard is common to each of the plurality of software developers (col. 14:20-25 “... results may be gathered after generation ...” and col. 11:30-35 “... produce multiple test cases ...”).

Art Unit: 2192

Per claim 13

Jorapur discloses:

The method of claim 10, wherein the set standard is unique to at least one of the plurality of software developers (col. 14:20-25 "... results may be gathered after generation ..." and col. 11:30-35 "... produce multiple test cases ...").

Per claim 14 (Currently Amended)

This is the system version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 15

This is the system version of the claimed method discussed above (Claim 2), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 16

This is the system version of the claimed method discussed above (Claim 3), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Art Unit: 2192

Per claim 17

This is the system version of the claimed method discussed above (Claim 4), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 18

This is the system version of the claimed method discussed above (Claim 5), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 20

This is the system version of the claimed method discussed above (Claim 7), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 21

This is the system version of the claimed method discussed above (Claim 8), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Art Unit: 2192

Per claim 22

This is the system version of the claimed method discussed above (Claim 9), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 23

This is the system version of the claimed method discussed above (Claim 10), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 24

This is the system version of the claimed method discussed above (Claim 11), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 26

This is the system version of the claimed method discussed above (Claim 13), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 27 (Currently Amended)

This is another method version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 28

This is another method version of the claimed method discussed above (Claim 3), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 29

This is another method version of the claimed method discussed above (Claim 4), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 30

This is another method version of the claimed method discussed above (Claim 5), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Jorapur.

Per claim 43 (Currently Amended)

Jorapur discloses:

The method of claim 28 further comprising customizing the verification scope of one or more of the plurality of verification ~~tools~~ programs for a second time, if the known error is not detected by executing the plurality of software verification ~~tools~~ programs (col. 4:55-65 “... each test may be generated in one or more blocks corresponding to one or more parts of the application to be tested ... “ and col. 6: 52-60 “... tests 302 generated may include test code that may be inserted ...” and col. 9: 45-55 “... test may be executed to assess the operation and function of the application to be tested ...”and e.g. FIG. 4, step 408 and related text).

Per claim 44 (Currently Amended)

Jorapur discloses:

The method of claim 27 further comprising executing the plurality of software verification ~~tools~~ programs periodically to prevent the known error from re-occurring when the computer software is modified (e.g. FIG. 4 and related text).

Per claim 45 (Currently Amended)

Jorapur discloses:

A system for automatically preventing errors in computer software having a plurality of different life cycle phases comprising:

means for providing a known error in the computer software, the known error belonging to a class of errors (col. 14:20-25 "... results may be gathered after generation ..." and col. 11:30-35 "... produce multiple test cases ...");

means for providing a plurality of software verification tools each of the plurality of software verification ~~tools~~ programs related to a respective lifecycle phase of the computer software (col. 4:55-65 "... each test may be generated in one or more blocks corresponding to one or more parts of the application to be tested ... " and col. 6: 52-60 "... tests 302 generated may include test code that may be inserted ..." and col. 9: 45-55 "... test may be executed to assess the operation and function of the application to be tested ..." and e.g. FIG. 4, step 408 and related text);

means for analyzing the known error in the computer software ~~to determine~~

means for determining what phase of the lifecycle the error was introduced, ~~introduce~~ based on analyzing the known error; (e.g. FIG. 6, 603 and related text); and

means for customizing a verification scope of one or more of the plurality of verification tools that correspond to the lifecycle phase ~~that~~ wherein the known error was introduced (e.g. FIG. 3, Configurations 303 and related text).

means for executing the plurality of software verification ~~tools~~ programs to verify the class of the known error is detected in a respective lifecycle phase of the computer software.

Per claim 46 (Currently Amended)

Jorapur discloses:

The system of claim 45 further comprising means for executing the plurality of

Art Unit: 2192

software verification ~~tools~~ programs to verify the known error is detected in computer software (col. 4:55-65 “... each test may be generated in one or more blocks corresponding to one or more parts of the application to be tested ... “ and col. 6: 52-60 “... tests 302 generated may include test code that may be inserted ...” and col. 9: 45-55 “... test may be executed to assess the operation and function of the application to be tested ...”and e.g. FIG. 4, step 408 and related text).

Per claim 47 (Currently Amended)

Jorapur discloses:

The system of claim 46 further comprising means for customizing the verification scope of one or more of the plurality of verification ~~tools~~ programs for a second time, if the known error is not detected by executing the plurality of software verification tools (col. 4:55-65 “... each test may be generated in one or more blocks corresponding to one or more parts of the application to be tested ... “ and col. 6: 52-60 “... tests 302 generated may include test code that may be inserted ...” and col. 9: 45-55 “... test may be executed to assess the operation and function of the application to be tested ...”and e.g. FIG. 4, step 408 and related text).

Per claim 48 (Currently Amended)

Jorapur discloses:

The system of claim 45 further comprising means for executing the plurality of

Art Unit: 2192

software verification ~~tools~~ programs periodically to prevent the known error from re-occurring when the computer software is modified (e.g. FIG. 4 and related text).

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 6, 19 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jorapur (US 7,299,382 B2) in view of Man et al. (US 6,625,760 B1).

Per claim 6 (Currently Amended)

Jorapur discloses:

The method of claim 1, further comprising formulating the verification results in a confidence factor represented by the equation: $C = p/t \times 100$, where p is number of successful test cases and t is total number of test cases.

Jorapur does not explicitly disclose formulating the verification results in a confidence factor represented by the equation above. However, Man discloses a significant test case is one

Art Unit: 2192

which has a high potential to uncover the presence of an error. Thus, successful execution of a significant test cases increases the programmer's confidence of the correctness of the program (emphasis supplied). Therefore it would have been obvious to one ordinary skilled in the art at the time the invention was made to formulate the verification results in a ration of number of successful test cases and total number of test cases to run a large number of test cases, a number of significant test cases that are representative of all the possible test conditions so that one can then rely more on the super set of another significant test set. On the other hand, since testing is costly in both man-hours and machine-time, it is the object of the programmer to limit the number of possible experiments such as the above ration as suggested by Man once in col. 1:55-67).

Per claim 19

This is the system version of the claimed method discussed above (Claim 6), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 31

This is another method version of the claimed method discussed above (Claim 6), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Conclusion

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ISAAC T. TECKLU whose telephone number is (571)272-7957. The examiner can normally be reached on M-TH 9:300A - 8:00P.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Isaac T Tecklu/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192